

PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2014-2015

Prova scritta del 15/07/2015

Scrivere **IN STAMPATELLO** COGNOME, NOME, MATRICOLA e CORSO su ogni foglio consegnato

ESERCIZIO 1 (6 punti)

Scrivere una grammatica **regolare** che genera il seguente linguaggio sull'alfabeto $\Lambda = \{a, b, c, d\}$

$$\mathcal{L} = \{x\alpha x \mid x \in \Lambda \wedge \alpha \in \Lambda^*\}$$

ESERCIZIO 2 (6 punti)

Dato un albero binario, si definisce *livello* di un nodo nell'albero il numero di nodi che si incontrano nel cammino dalla radice al nodo medesimo. Quindi, ad esempio, la radice ha livello 1, i figli della radice hanno livello 2, e così via. Dato il tipo degli alberi binari visti a lezione

```
type 'a btree = Void | Node of 'a * 'a btree * 'a btree
```

si scriva in CAML una funzione

```
foo : 'a btree -> int -> 'a list * 'a list
```

che, dato un albero binario e un intero positivo n , restituisce una coppia di liste $(l1, l2)$ in cui $l1$ contiene tutti e soli i valori dei nodi che occorrono a livelli dell'albero minori o uguali a n e $l2$ contiene tutti e soli i valori degli altri nodi dell'albero.

ESERCIZIO 3 (6 punti)

Si definisca in CAML, senza utilizzare ricorsione esplicita, una funzione

```
twice : 'a -> 'a list -> bool
```

in modo che `(twice x xs)` restituisca `true` se `x` occorre in `xs` esattamente due volte, restituisca `false` altrimenti.

ESERCIZIO 4 (6 punti)

Si scriva in C una procedura che, prese attraverso opportuni parametri due liste di interi, elimina dalla seconda lista tutti gli elementi che non occorrono anche nella prima.

Si suppongano predefiniti i tipi

```
struct el {int info; struct el *next;};
```

```
typedef struct el ElementoDiLista;  
typedef ElementoDiLista *ListaDiInteri;
```

ESERCIZIO 5 (6 punti)

Si scriva in C una funzione che, dati due array di interi `a` e `b` e le loro dimensioni `dima` e `dimb`, restituisce il valore di verità della seguente formula

$$\exists i \in [0, dima). \#\{j \mid j \in [0, dimb) \wedge a[i] = b[j]\} > \#\{k \mid k \in [0, dima) \wedge a[i] = a[k]\}$$

Si ricorda che dato un insieme finito A , $\#A$ indica il numero di elementi di A .